Scientia Iranica, Vol. 9, No. 3, pp 255–262

SCIENTIA
IRANICA

Using Neural Network and Genetic Algorithms to Solve a Multiple Attributes Knapsack Problem

M. Ghazanfari* and M. Nojavan[1]

Selecting an optimum combination of items from a set of items, known as knapsack problems, is an important issue in the decision making domain. In this paper, a new approach is developed to solve a Multiple Attribute Knapsack Problem (MAKP) in which each combination of items is evaluated using some quantitative and qualitative attributes. The assumed qualitative attributes cannot be measured by a mathematical formulation but by a DM/expert. In this paper, a Genetic Algorithm (GA) model has been developed to generate different combinations as the sequential population of the GA model. To rate the qualitative attributes for each chromosome (or combination) in the population, a Neural Network (NN) model has been developed. The ratings (or scores) resulted from quantitative attributes (by NN) and qualitative attributes (by mathematical formulation) for each chromosome form a row of a decision matrix. Having the decision matrix and known weights of attributes, the combinations in each population are ranked by applying a MADM model. The ranks obtained for each chromosome shows the fitness of that chromosome. Using the GA model, the best combination is achieved. The results of conducted experiments show the capability of the proposed approach to deal with MAKP problems.

INTRODUCTION

Selecting an optimum combination from a set of items, known as knapsack problems, is an important issue in the decision making domain. It is usually applied to solve problems such as capital budgeting, cargo loading, cutting stock, portfolio planning etc. A knapsack problem is usually involved in a single objective or a group of objectives, subject to some constraints [1,2]. A classical knapsack problem can be mathematically formulated as follows:

$$\max \sum_{j=1}^{m} p_j x_j$$

$$\text{s.t.} \quad \sum_{j=1}^{m} b_j x_j \leq B$$

$$x_j = \begin{cases} 1, & \text{if item } j \text{ is selected} \\ 0, & \text{otherwise,} \end{cases}$$

where m is the number of items, P_j the profit of item $j(j = 1, ..., m)$, B the available resource and b_j the amount of resource used by item j.

There are many variations of the knapsack problem in the literature, such as the multiple choice, the bounded and the unbounded knapsack problems. It has been proven that 0-1 knapsack problems are NP-hard [3]. Several researchers have solved the knapsack problem using genetic algorithms [3-5]. Moreover some researchers have solved multi-objectives knapsack problems with crisp objective [6]. The multi-objectives knapsack problem has been also solved in fuzzy environment [7].

In general, a knapsack problem consists of multiple objectives and linear or nonlinear constraints. In this case, the knapsack problem can be formulated as follows:

$$\text{Optimum } \{f_1(x), ..., f_k(x)\}$$

$$\text{s.t.} \quad g_i(x) \leq B_i \quad (i = 1, ..., l),$$

$$x = (x_1, ..., x_m), \quad x \in \{0, 2\},$$

in which, k is the number of objectives, $f_j(x)$ the jth objective function $(j = 1, ..., k)$, l the number of resources available, $g_i(x)$ the ith constraint $(i = 1, ..., l)$ and B_i the ith resource.

*. *Corresponding Author, Department of Industrial Engineering, the University of Science & Technology, Tehran, I.R. Iran.*

1. *Faculty of Technology, Science and Research Campus, Islamic Azad University, Tehran, I.R. Iran.*

In this paper, a new method has been developed to solve a multiple objective knapsack problem, in which each combination of items is evaluated using some quantitative and qualitative objectives. It is also assumed that the qualitative objectives could not be measured by a numerical mathematical function. Sometimes the opinion of the DM/expert is mainly subjective and may not also be explained by a set of rules. "Combination Consistency" and "Combination Diversity" are two examples of such objectives, which cannot be stated by a measurable function. These objectives are rated based on the expert's opinion.

It is not possible to use conventional approaches in this case and application of a Multi-Attribute Decision Making (MADM) model is required. This problem is called a Multiple Attributes Knapsack Problem (MAKP) and an integrated procedure is proposed to solve it.

PROBLEM DESCRIPTION

Consider a Multiple Attributes Knapsack Problem (MAKP), in which each combination of items is evaluated by k quantitative and qualitative objectives. The ratings of quantitative objectives are measured using mathematical formulations. However, the ratings (or scores) for qualitative objectives are set by an expert. Considering each combination of items as an alternative and each objective as an attribute, one can interpret the MAKP as a MADM problem, as shown in Figure 1.

Although the matrix shown in Figure 1 has 2^m rows (equal to the number of possible combinations), some of the combinations are infeasible regarding the constraints and, therefore, are discarded from further consideration.

After filtering these combinations from the decision matrix, both quantitative and qualitative objectives are evaluated for each feasible combination. The former is evaluated using Mathematical Formulation (MF) and the latter is evaluated based on expert/DM opinion.

Objective / Combination	f_1	f_k
A_1	$r_{1,1}$	$r_{1,k}$
$\vdots$	$\vdots$	$\vdots$
A_{2^m}	$r_{2^m,1}$	$r_{2^m,k}$

Figure 1. Converting a MAKP to a MADM model. m is the number of items (or objects), A_i the ith combination (or alternative)$(i = 1, ..., 2^m)$, k the number of objectives (or attributes), f_j the jth objective function (or attribute) $(j = 1, ..., k)$ and r_{ij} the rating for objective function j in combination i.

In this stage, if the weights or relative importance of objective functions are known, one of the conventional MADM models, such as SAW, TOPSIS or others [8] can be used to rank the combinations (or alternatives). The combination with the highest rank is selected as the optimum solution. Since this approach continuously uses the expert/DM opinion, it is known as the "Interactive-MADM" or "MADM-expert" approach.

It is very difficult to apply the "MADM-expert" approach to solve the MAKP when the number of items in each combination (m) is high. This is because searching among 2^m combinations to find the optimum solution is very time consuming and eventually impractical. To solve this difficulty, a Genetic Algorithms (GA) model has been applied. The GA can effectively search the space which consists of a huge number of alternatives [9-11].

The GA model sequentially forms populations, using the chromosomes that are a combination of items. Following the standard steps of GA, the (near) optimum solution can be found. Since the values or ratings of qualitative objectives during the GA process need to be specified by an expert/DM, this approach is known as the "Interactive-GA" or "GA-expert" approach.

There is still another difficulty in solving MAKP using the GA-expert approach. Due to the huge number of combinations (chromosomes), the evaluation of qualitative objective functions is a hard and erratic task for the DM/expert.

To solve the aforementioned problem, a Neural Network (NN) model has been proposed. NN models are for applying complicated functions without specifying their detailed structures [12-14]. The supervised NN models require training during the creating process. For this purpose, a list of samples are requested from the DM/expert to train the NN model for evaluating qualitative objectives. The DM/expert opinion is replaced by the trained NN model.

The resulting approach to solving MAKP is integrated and called the "hybrid GA-NN" approach. This framework of approach is built by the GA model. The solution process of MAKP by the "hybrid GA-NN" approach has three stages as follows:

1. In the first stage, a population is formed from feasible combinations by the proposed GA model;

2. In the second stage, the quantitative and qualitative objective functions are evaluated for each combination. The former is accomplished by Mathematical Formulation (MF) and the latter by the trained NN model;

3. In the third stage, all combinations (chromosomes in the population) are ranked using a MADM model.

The rank resulting from the last stage is used for selecting the new population and the process is iterated to find the final solution. Figure 2 shows the integrated approach to solve MAKP by the "hybrid GA-NN" model. The solutions are explained in detail in the following.

First Stage: Generation and Selection

In this stage, the GA method is used to generate and select the combinations (chromosomes). The GA model developed in this paper works as follows.

Initial Population

The initial population consists of n chromosomes. Each chromosome has m genes with value of 0 or 1. The initial population is generated randomly.

Crossover Operation

Regarding crossover probability (P_c), each chromosome is selected for crossover. The crossover operator uses two chromosomes (parents) and generates two new chromosomes (offspring). The Single Point Cut (SPC) method is used as the crossover operator, i.e., a point between 1 to $m-1$ is selected randomly for both chromosomes and, then, the right side of the cross point are two parents exchanged to create two offspring.

Mutation Operation

Based on mutation probability (P_m), the value of a gene in a chromosome is changed from 0 to 1 or vice versa.

Feasibility Test

The chromosomes are checked against all constraints. There are three major strategies to deal with the infeasible solution, i.e., rejecting, repairing and penalizing. In the first strategy, all infeasible chromosomes are discarded from further consideration. The repairing strategy involves taking an infeasible chromosome and generating a feasible one through some repairing procedure. The last strategy accepts the infeasible solutions but penalizes them and decreases their fitness corresponding to their infeasibilities.

Due to the simplicity of repair strategy, this paper uses this strategy. If the chromosome (a combination of items) violates any constraint, an item is eliminated from (or added to) the combination by changing a 1 to 0 (or changing a 0 to 1). This step is continued until all constraints are satisfied. The gene for the change is

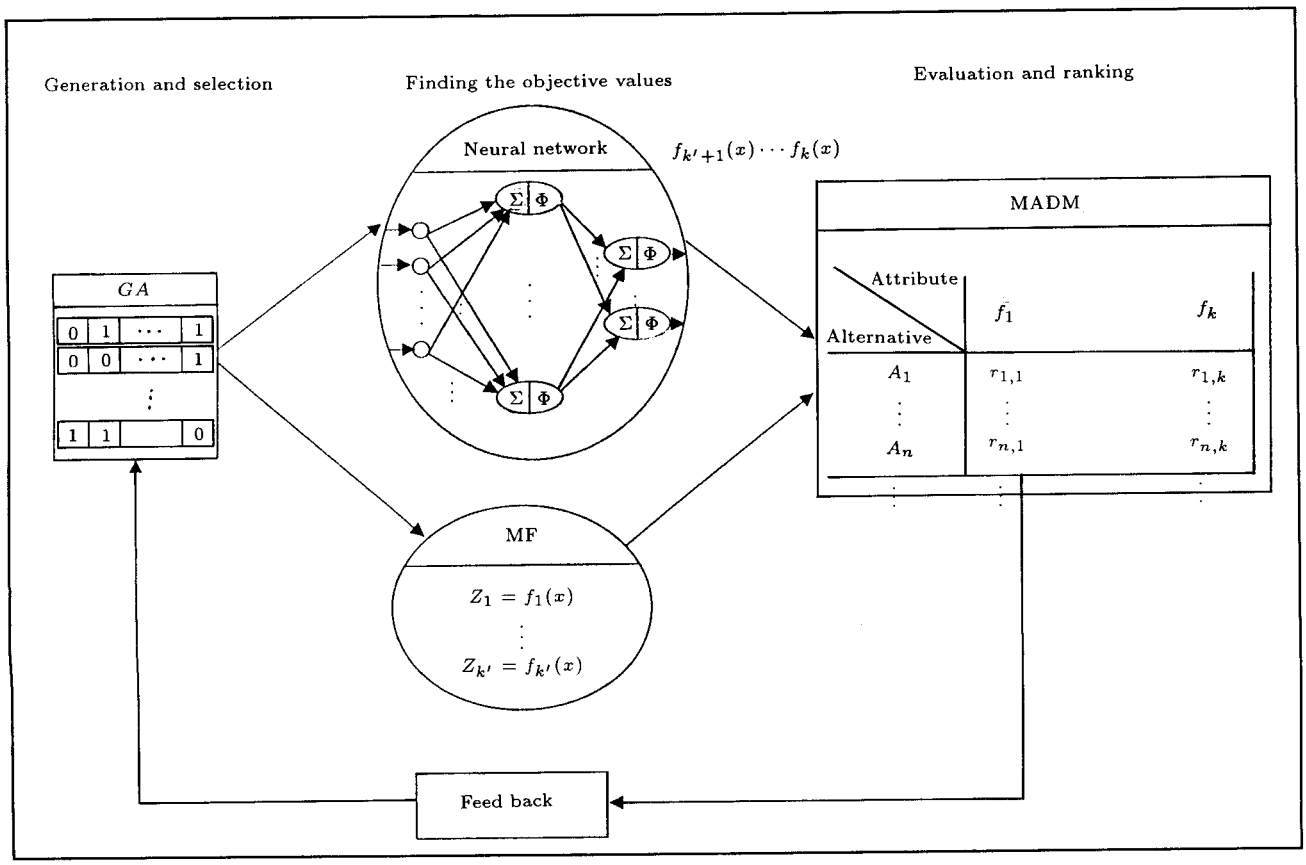

Figure 2. The integrated approach for solving MAKP.

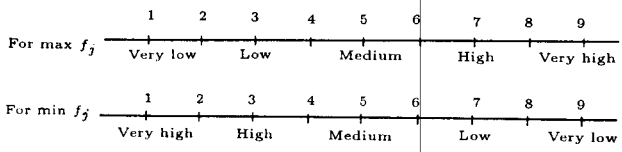

Figure 3. The ratings for qualitative attributes.

selected randomly. To be more efficient, in this paper a heuristic for the changing process has been applied. The selection of the gene for change is based on its benefit/cost fraction.

Sampling Space

The sampling space is specified based on the selection method. If the selection method is probabilistic, the parents are replaced by offspring and the selection procedure is fulfilled. When the selection method is deterministic, the offspring are added to the population and then the selection is fulfilled.

Selection Operation

The new population is formed based on the selection method. If the selection method is probabilistic, the roulette wheel is applied. In this case, the normalized rank of each chromosome from the third stage is used as the selective probability for that chromosome. For the deterministic approach, the parents and offspring are added together and sorted by their ranks. Then n first chromosomes are selected from the top of the list.

Second Stage: Finding the Objective Values

In this stage, the values of all objectives are determined to form the decision matrix completely. In other word, the values (or ratings) are required, both for quantitative and qualitative objectives (attributes), for all chromosomes in the population. The ratings for the first group of attributes are calculated based on mathematical formulations, which are known already. The ratings for the second group are determined by the trained NN model.

The NN model developed in this paper has the following structure.

Training Set Determination

A complete training set is provided covering different combinations. The data for each sample is entered as a binary vector. The dimension of the vector is m. The ratings for qualitative attributes in each combination are determined by the DM/expert, using one of the linguistic variables shown in Figure 3.

Neural Network Design

The number of input nodes are equal to the number of items in each combination (i.e., m). The desired output for each attribute should be an integer number between 1 to 9. However, the output of each node in the NN model is between 0 to 1. Therefore, the desired output was changed to be between 0 to 1010 in a binary mode and a quadruple output node is used to generate these binary numbers. Regarding the above description, the output nodes are 4 times the qualitative attributes. The middle (hidden) layer is to keep the knowledge and help the convergence. A maximum number of hidden layer nodes are 2 times the input nodes plus one [14]. Therefore, the maximum number is $2m + 1$. A bias term is also used in the architecture of the proposed NN model. This term is a weight on a connection that has its input value always equal to 1. A bias term sometimes helps the convergence of the weights to an acceptable solution and its use is largely a matter of experimentation with the specific application [11,14].

Initialization

Weights (on hidden and output layers) should be initialized to small, random values, say between ± 0.5 as should the bias terms.

Error Calculation and Weight Updating

Summing up the values in middle and output nodes for each sample and applying the sigmoid function, outputs of the NN model are calculated. These outputs are compared with the binary format of the DM/expert opinion about qualitative attributes for that sample. Then, the Back Propagation Net (BPN) learning method is used to update the weights. After using all samples in the training set, Mean Squared Error (MSE) is calculated for them as follows:

$$MSE = \sum_{j=1}^{n\_samples} \varepsilon_j^2/(n\_samples), \qquad (1)$$

where ε_j is the error of sample j in training set and $n\_samples$ is the number of samples in the training set.

Termination Test

If the MSE result is equal to, or less than, the predetermined value, or the number iterations reaches a predetermined value, the training is stopped, otherwise the training is continued.

Third Stage: Evaluation and Ranking

So far the values of both quantitative and qualitative objectives have been assessed for all chromosomes of the population. These values form a MADM decision matrix in which chromosomes are alternatives, objectives are attributes and aforementioned values are the ratings of each alternative against the attributes. Applying the relative weights of objectives, this MADM problem can be solved using one of the conventional models such as SAW or TOPSIS.

SAW has been applied to rank the different chromosomes in each population. SAW requires the elements of decision matrix to be normalized. The following equations are used for normalization.

$$\bar{r}_{ij} = r_{ij} / \sum_{i=1}^{n} r_{ij},$$

if objective j has to be maximized,

$$\bar{r}_{ij} = (1/r_{ij}) / \sum_{i=1}^{n} (1/r_{ij}),$$

if objective j has to be minimized, (2)

in which $\bar{r}_{ij}$ is the normalized ratings of combination i for attribute j.

In this case, the rank for each chromosome is calculated as follows:

$$P_i = \sum_{j=1}^{k} w_j \bar{r}_{ij}, \quad (3)$$

in which P_i is the rank of combination i and w_j is relative weight of attribute j. Considering the ranks obtained in the third stage as the fitness of each chromosome, the generation of a new population is undertaken using the process explained in the first stage.

This 3-stage procedure is iterated until a stop condition is satisfied, that is a converge solution or the pre-determined number of iterations. At the end of the procedure, the solutions are ranked and the solution with the highest rank is selected as the best.

The pseudocode of the proposed procedure is shown in Figure 4.

COMPUTATIONAL EXPERIMENT

All approaches proposed in this paper were coded in FORTRAN 90. To check the ability of procedure to solve a multi-attributes knapsack problem, a number of examples were conducted. Due to lack of such models in the literature, authors could not give any comparison. Given a typical example, the results of computations based on the proposed model are indicated.

Example

Assume the best combination of projects in a project investment decision problem is sought. There are two goals. The first goal, maximizing the profit from investment, is a quantitative one. The second goal, maximizing the consistency of the selected projects, is a qualitative one. By consistency, it is meant that the selected projects belong to a similar branch of industry. In other words, the projects from a more or less similar family of specialization can be managed more easily.

In the example, there are 4 groups of projects, each of which are completely similar and, therefore, consistent. The degree of similarity or consistency of other combinations or projects is determined by the DM/ expert. Table 1 shows the projects, the profit and the cost of available projects.

There are three constraints in this example. The first is a constraint on the total available budget, which is B for investment. The second one is limitation on the number of projects to be selected. The last one is the lower level on the consistency of the solution. The mathematical programming model for this example is as follows:

$$W_1 : \max Z_1 =$$

$$\{20x_1 + 10x_2 + 15x_3 + 17x_4 + 19x_5 + 11x_6 + 13x_7 + 7x_8 + 22x_9 + 14x_{10}\}$$

$$W_2 : \max Z_2 =$$

$$\{\text{Consistency}(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})\}$$

$$\text{s.t.} \begin{cases} 100x_1 + 60x_2 + 90x_3 + 95x_4 + 50x_5 + 100x_6 + 70x_7 + 20x_8 + 90x_9 + 80x_{10} \le B \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} \le 8 \\ \text{Consistency}(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}) \ge 3 \\ x_j = \{0, 1\} \end{cases}$$

To solve the problem, three stages must be passed based on the proposed procedure. Before starting the solution process, a NN model is developed. The specifications of the NN model for this example are as follows:

- Input nodes: 10 (plus one node for bias),
- Hidden nodes: 5 (plus one node for bias),
- Output nodes: 4,
- Learning method: Back Propagation Net (BPN),
- Input process: dot product,
- Network type: Multiple layers, feed forward,

Table 1. The specifications of projects.

Category	1		2			3		4		
Project	1	2	3	4	5	6	7	8	9	10
Profit (annual)	20	10	15	17	19	11	13	7	22	14
Cost (investment)	100	60	90	95	50	100	70	20	90	80

Pre-step 1. **Neural Network Design**
number of iterations: max-iter
number of samples in training set: $n-$samples
number of nodes in hidden layer: $n-$hidden
number of nodes in output layer: $n-$output
activation function: Sigmoid
learning method: Back Propagation Net (BPN)
minimum of mean squared error: $min-$MSE
slope parameter: α
number of items: m
current iteration: iter $\leftarrow$

Pre-step 2. Training Set Determination
for $k \leftarrow 1$ **to** $n-$samples **do**
use DM opinion to specify combination k and the ratings of it's qualitative attributes
end

Pre-step 3. Initialization
create the weights of hidden, output and bias nodes by random (-0.5, 0.5)

Pre-step 4. Errors Calculation and Weights Updating
for $k \leftarrow 1$ **to** $n-$samples **do**
compute the output ratings from the NN model
compare the output ratings by the NN and desired ratings by DM
update the weights of NN model by the BPN method
end

Pre-step 5. Termination Test
iter $\leftarrow$ iter +1
if (iter $\geq$ max-iter) **then**
stop
else
compute Mean Squared Error (MSE) for training set
if (MSE $<$ $min-$MSE) **then**
stop
else
go to pre-step 4
end
end

Step 1. GA Specification
number of generations : max-gen
number of offsprings : $n-$offspring
number of genes in each chromosome: m
population size: pop-size
probability of crossover: P_c
probability of mutation: P_m
current generation: gene $\leftarrow$ 0

Step 2. Initial Population
for $k \leftarrow 1$ **to** pop-size **do**
creat combination (chromosome) k randomly
end

Step 3. Crossover Operation
$n-$offspring $\leftarrow$ 0
for $k \leftarrow 1$ **to** pop-size / 2 **do**
if (random( ) $\leq P_c$) **then**
$j \leftarrow$ random (pop-size)
$i \leftarrow$ random (pop-size)
cross point $\leftarrow$ random$(m-1)$
create two offspring using the single point cut (SPC) method
$n-$offspring $\leftarrow$ $n-$offspring + 2
end
end

Step 4. Mutation Operation
for $k \leftarrow 1$ **to** pop-size **do**
for l $\leftarrow$ 1 **to** m **do**
if (random( ) $\leq P_m$) **then**
save the number of mutated genes in chromosome k
end
end
create offspring by changing the mutated genes from 1 to 0 and vise versa
$n-$offspring $\leftarrow$ $n-$offspring + 1
end

Step 5. Feasibility Test
for $k \leftarrow 1$ **to** $n-$offspring **do**
if (offspring k is infeasible) **then**
repair (offspring k)
end
end

Step 6. Sampling Space Determination
if (selection is deterministic) then
for $k \leftarrow 1$ **to** $n-$offspring **do**
insert offspring k to population
end
end
if (selection is probabilistic) **then**
for $k \leftarrow 1$ **to** $n-$offspring **do**
replace offspring k to its parent
end
end

Step 7. Population Evaluation
if (selection is deterministic) **then**
for $k \leftarrow 1$ **to** (pop-size+n)-(offspring) **do**
compute the ratings of qualitative attributes for chromosome k using the NN
compute the ratings of quantitative attributes for chromosome k using mathematical formulation
end
for $k \leftarrow 1$ **to** (pop-size+n)-(offspring) **do**
rank chromosome k using decision matrix, the weights of attributes and the MADM model
end
end
if (selection is probabilistic) **then**
for $k \leftarrow 1$ **to** pop-size **do**
compute the ratings of qualitative attributes for chromosome k using the NN
compute the ratings of quantitative attributes for chromosome k using mathematical formulation
end
for $k \leftarrow 1$ **to** pop-size **do**
compute the rank of chromosome k using the decision matrix, the weights of attributes and the MADM model
end
end

Step 8. Selection Operation
if (selection is deterministic) **then**
sort chromosomes in the population based on their ranks
select pop-size chromosomes from the top of population
end
if (selection is probabilistic) **then**
for $k \leftarrow 1$ **to** pop-size **do**
normalize the rank of chromosome k
end
for $k \leftarrow 1$ **to** pop-size **do**
compute selective probabilities p_k = rank(chromosome k)
compute cumulative probabilities $q_k = \sum_{j=1}^{k} p_j$
end
$q_0 \leftarrow 0$
for $k \leftarrow 1$ to pop-size **do**
if ($q_{k-1} <$ random( ) $\leq q_k$) **then**
select chromosome k
end
end
end

Step 9. Termination Test
gen $\leftarrow$ gen +1
if (gen $<$ max-gen) **then**
go to Step 3
else
stop
end

Figure 4. The pseudocode of the proposed procedure; note random ( ) means to return a random real number in (0,1) and random (num) means to return a random integer number in [1, num].

- Activation function (in hidden and output nodes): Sigmoid,
- Learning- rate parameter = 0.01,
- Slope parameter =1.

A sample has been applied, consisting of 40 combinations, to train the neural net model. The rating for consistency of each combination was specified by the DM using the interval scale shown in Figure 5.

After 1785 iterations, the model was trained with the Mean Squared Error (MSE), less 0.05. To solve the problem, the GA model was run with the following specifications:

- Probability of crossover: $(0.5 \leq P_c \leq 0.7)$,
- Probability of mutation: $(0.02 \leq P_m \leq 0.15)$,
- Population size: $(10 \leq n \leq 20)$,
- Termination condition: $(20 \leq t_{\max} \leq 50)$,
- Selection type = deterministic,
- The kind of MADM model = SAW.

The problem is solved with different B (budget available) and W (the relative weights of objectives). Analyzing the experiments conducted shows that the following values for GA parameters are more appropriate: $P_c = 0.6$, $P_m = 0.1$, $n = 10$ and $t_{\max} = 50$.

Using enumeration method, it is also clear that the results of the proposed procedure are optimum for a small budget. The final solution shown in Table 2 is sensitive to the values of objectives weights.

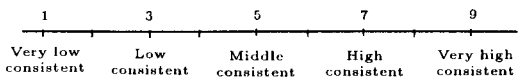

Figure 5. The ratings for consistency of combinations.

CONCLUSION

Decision making is involved in multiple, usually conflicting, objectives or attributes. The knapsack problem, selecting an optimum combination of items from a set of these items, is a classical decision making model with a single objective and constraint. Extending this model, some researchers have developed models involving multiple quantitative objectives and constraints and solved them using Multi-Objective Decision Making (MODM) models. This paper extends the concept of knapsack model to deal with a situation in which multiple (quantitative and qualitative) attributes are influencing the final decision. The MADM model is used to solve this problem. The proposed approach integrates three domains, i.e, neural network, genetic algorithms and the MADM model.

The MADM model is applied to rank alternatives, based on their ratings for both quantitative and qualitative attributes. Each alternative resembles a feasible combination of items from the set of available items.

Since the number of possible combinations is 2^m, the Genetic Algorithms method is used to search the space of solution effectively. Evaluating the rating

Table 2. The results of solutions with different B and W.

		W=(0.6, 0.4)				W=(0.8, 0.2)			
No	B	Final Solution	Z_1	Z_2	B^*	Final Solution	Z_1	Z_2	B^*
1	50	{5}	19	9	50	{5}	19	9	50
2	100	{9}	22	9	90	{9}	22	9	90
3	150	{4,5}	36	9	145	{5,9}	41	6	140
4	200	{8,9,10}	43	9	190	{5,8,9}	48	6	160
5	250	{3,4,5}	51	9	235	{3,5,8,9}	63	6	240
6	300	{3,4,5}	51	9	235	{4,5,8,9}	65	6	255
7	350	{3,4,5,8,9}	80	6	345	{3,4,5,8,9}	80	6	345
8	400	{1,2,3,4,5}	81	6	395	{1,2,3,4,5}	81	6	395
9	450	{3,4,5,8,9,10}	94	6	425	{3,4,5,8,9,10}	94	6	425
10	500	{3,4,5,8,9,10}	94	6	425	{3,4,5,8,9,10}	94	6	425
11	550	{3,4,5,8,9,10}	94	6	425	{1,3,4,5,8,9,10}	114	3	525
12	600	{3,4,5,8,9,10}	94	6	425	{1,2,3,4,5,8,9,10}	124	3	585
13	650	{3,4,5,8,9,10}	94	6	425	{1,2,3,4,5,8,9,10}	124	3	585
14	700	{3,4,5,8,9,10}	94	6	425	{1,2,3,4,5,8,9,10}	124	3	585
15	750	{3,4,5,8,9,10}	94	6	425	{1,2,3,4,5,8,9,10}	124	3	585

B^* is the actual budget used by the projects selected.

of attributes for each combination is a skilled job for a DM and cannot be done easily because of the present qualitative attributes. It is also difficult for a DM, due to the number of combinations available for evaluation. To solve these difficulties, the DM was replaced by an NN model. That is, using the samples in the training set, the NN model was trained and applied to find the ratings for all qualitative attributes in each chromosome or combination. These ratings and ratings from the mathematical functions for quantitative attributes, form a row of decision matrix. The fitness for each chromosome is calculated, based on the results of the applied MADM model to rank chromosomes, which is, in fact, the final rating for each combination. In other words, a MADM model (such as SAW) integrates attributes for each chromosome, using the individual ratings and weight of each attribute.

The results of the conducted experiments show that the proposed approach is a promising method for solving the multiple-attributes knapsack problem (MAKP), based on quantitative and qualitative objectives.

Solving the knapsack problem in a fuzzy environment and applying expert systems in place of the NN model can be regarded as further areas for research.

REFERENCES

1. Marty, K., *Linear and Combinatorial Programming*, John Wiley & Sons, New York, USA (1978).
2. Martello, S. and Toth, P., *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley & Sons, Chichester (1990).
3. Gordon, V. and Whitney, D. "Serial and parallel genetic algorithms as functions, optimizers", *Proceedings of the Fifth International Conference on GA*, Morgan Kaufmonn Publishers, Forrest, S., Ed., San Mateo, CA, pp 177-133 (1993).
4. Hinterding, R. "Mapping order-independent genes and the knapsack problem", *Proceedings of the First IEEE Conference on Evolutionary Computations*, IEEE Press, Fogel, D., Ed., Orlando, FL, pp 13-17 (1994).
5. Olsen, A. "Penalty function and the knapsack problem", *Proceedings of the First IEEE Conference on Evolutionary Computations*, IEEE Press, Fogel, D., Ed., Orlando, FL, pp 554-558 (1994).
6. Fonseca, C.M. and Fleming, P.J. "An overview of evolutionary algorithms in multi-objective optimisation", *Evolutionary Computation*, **3**, pp 1-16 (Spring, 1995).
7. Sakawa, M., Kato, K. and Shibano, T. "Fuzzy programming for multi-objectives 0-1 programming problem through revised genetic algorithms", *European Journal of Operational Research* (1998).
8. Hwang, C.L. and Yoon, K., *Multiple Attributes Decision Making*, Springer Verlag (1991).
9. Reeves, C.R., Ed., *Modern Heuristic Techniques for Combinational Problem*, Chapter 4, Blackwell Scientific, England (1993).
10. Chu, P.C. and Beasley, J.E. "A genetic algorithm for the generalized assignment problem", *Comp. Opns. Res*, **24**, pp 17-23 (1997).
11. Holland, J.H., *Adaptation in Natural and Artificial Systems*, Michigan Press University, Ann Arbor, MI, (1975).
12. Freeman, J.A. and Sakpura, D.M., *Neural Networks: Algorithms, Applications and Programming Techniques*, Reading, MA: Adisson-Wesly, USA (1991).
13. Haykin, S., *Neural Network: A Comprehensive Foundation*, Macmillan College Publishing Co., N.Y., USA (1994).
14. Sejnowski, T.J., Kienker, B.K. and Hinton, G.E. "Learning symmetry groups with hidden units: Beyond the perception", *Physica*, **22D**, pp 260-275 (1986).